The Chiba City Project:
Advanced Scalability and Development Testbeds

Rick Stevens and Rémy Evard
stevens@mcs.anl.gov, evard@mcs.anl.gov
December 17, 2001

1 Introduction

One of the major research goals in computer science for the past 10 years has been the design of software and hardware systems that can scale over many orders of magnitude in systems performance. This goal, which is likely to be pursued for the next 10 years as well, is driven by the need to reduce the number of platforms that applications target, while enabling applications to scale from small model problems that fit on personal computing resources to full-scale simulations at the limit of systems we can build and deploy. Much research during the past decade has demonstrated that the primary barrier to achieving systems scalability is scalability of systems software (e.g., the software needed to manage and operate the systems and to support applications). Researchers investigating paths to petaflops-capable systems in the early and mid-1990s identified multiple possible hardware technology paths to petascale performance. Each of these hardware paths, regardless of the technology base, had one thing in common: the need for large degrees of concurrency in future systems. Little's law ($c = bw \times l$) relates concurrency to bandwidth and latency, where bandwidth is proportional to floating-point performance. It suggests that all future systems, whether they target teraflops, petaflops, or higher, will need to support increased concurrency even as bandwidth increases, because latency is fixed by the speed of light and by the process technology used for the components. Scalability is therefore a fundamental goal in high-performance systems research.

The objective of the Chiba City Project at Argonne is to provide a series of parallel hardware and software testbeds for the computer science and applications community aimed at supporting research in software scalability. The work supported by these testbeds will have broad impact on the high-performance computing (HPC) community by enabling more rapid progress in realizing the dream of scalable systems. The concept behind these testbeds is based on two critical issues that are limiting the rate of progress toward scalable systems.

First, no large HPC testbeds are available to support research of scalability problems in computer science. Nor, despite a recognition of the importance of reliable and robust systems software, are any large-scale testbeds available for the development and testing of systems software. As a community, we will soon be building systems for which we do not have the ability to research, develop, and test software effectively. This situation will affect software of all types: operating systems, systems software, middleware, libraries, and applications.

Second, cluster architectures, which are currently quite popular, are not likely to scale to petaflops in the pure commodity space. Therefore, the first petaflops systems, which are

expected to be feasible within six years, will most likely build on innovations in cluster based systems technology, CPU microarchitectures, and custom-designed memory and processor integration. These computers likely will have millions of individual systems components that must be managed by systems software. Current systems software can handle, at best, thousands of system elements. Thus, in the next five years, systems software must increase in capability by *at least three orders of magnitude*. Because we have no way today to test effectively at such a scale, very little of the activity necessary to meet these challenges is under way. At the current rate, it is exceedingly likely that when the hardware for petaflops becomes available, the software will be at least three years behind. As a result, future systems, in the best case, will need to undergo a lengthy test and tuning process before they can be released to the full user community — thus delaying critical scientific computational results. In the worst case, future systems will never reach the state of reliable production service at full scale.

Three years ago, we built Chiba City, a 256-node cluster, in order to serve as a preliminary scalability testbed. It has worked exceptionally well as a research platform and as a facilitator for systems software development. However, the scale of the system is limited.

Based on our testing and experience, we know that many software systems, for example schedulers, file systems, runtime libraries, and visualization tools, can now scale reasonably well to 256 nodes but are not likely to scale to 1024 or beyond without substantial new development and testing. Therefore we propose to continue the mission that started with the first system in the Chiba City project: the creation of an ongoing scalability testbed based on cluster technology, with regular hardware updates every two years. This testbed will specifically promote and support large-scale research and testing in computer science and selected areas of computational science. It will have the capacity to test on large numbers of "virtual" nodes, allowing scientists to push the scale of their research toward petaflops scalability barriers. Furthermore, it will support development and testing of systems software on real applications that is not yet ready for full deployment on production systems.

The system proposed here, to be installed in the first half of 2002, will have 1024 computing nodes and have a capacity of 8192 virtual nodes. Follow-on systems will be installed in 2004 and 2006, with the final system having a capacity for 160K virtual nodes providing a development and testing capability within the range needed for enabling petascale systems. Each successive scalability testbed in the project, like the first Chiba City system, will be available to computer scientists, software engineers, and application developers across all facets of the nation's high-performance computing community, including DOE-funded base program research activities, DOE SciDAC centers and projects, university-base researchers and companies contributing to the cluster software and technology development effort.

We are targeting a broad user community for this testbed and expect to have a significant population of users. Allocation of time on the testbed will be made in coordination with DOE's Office of Advanced Scientific Computing Research to meet program needs.

2 Purpose and User Community

The proposed project has two related goals; supporting scalability research, which we have discussed and motivated above, and enabling new modes of systems software development that require modifications to the deep software infrastructure of the system and testing with real applications. This later concept we call "extreme development" to distinguish it from routine

development that has less impact on the infrastructure of the system. Together, these two goals enable a higher-level objective: to support critical petaflops-enabling research and software development that would not otherwise be possible to carry out.

To achieve these goals, we will make the computing testbeds available to a wide range of researchers. Users will come from the following areas:

- DOE Base Program Research. A substantial number of existing research programs, including those supported by DOE base funding as well as collaborations with other institutions, require large-scale development, carry out scalability research, or need to be built on a system that supports extreme system development. Among these are MPICH; the Parallel Virtual File System; large-scale visualization research; ACTS, PETSc and other libraries; the Los Alamos Science Appliance project; the Programming Models project; and a number of specialized kernel development efforts in the DOE laboratories.
- DOE SciDAC Projects. A number of the DOE SciDAC projects must develop at scale and carry out extreme system development, including the Scalable Systems Software ISIC, the Scientific Data Management ISIC, and the Performance Evaluation Research Center.
- University Partners. Many computer scientists in universities need to test on large systems. A number of these researchers are already using Chiba City, but we will expand the reach and scope to include a much larger community. Based on requests thus far, we anticipate several hundred such researchers. Additionally, the system will be available to all DOE Early Career PIs in the areas of computer science and mathematics.
- Strategic Vendors. Various commercial entities are developing a key technology or software product that must perform well at scale in order to benefit the HPC community. Myricom, Scyld, and others have already used Chiba City for testing, to great benefit to the community. Other interested vendors include Veridian, TurboLinux, and Etnus. Additional vendors and third-party application developers are welcome and expected.

Argonne will organize several activities during FY02 to build user community for the scalability testbed.

1. We will conduct several workshops aimed at building a coherent user community for the system. These workshops will bring together DOE and University researchers to explore how to best use the testbed and how ANL can support the community.
2. We will build a web-based environment that will explain how to use the testbed and describe the current scalability goals and accomplishments for various projects that are using the testbed
3. We will develop a user support team at ANL that has familiarity with the various scalability research projects and that can provide focused assistance to the Chiba City user community that is developing systems software.

3 Significance

The goal of scalable HPC systems is at the core of multiple agency research programs in scientific computing (e.g., DOE OSC, DOE NNSA, NSF CISE, NASA HPC). The national HPC program and related efforts have all identified the need for scalable systems software as a prerequisite for scalable applications and software infrastructure that is affordable. Vendors have adopted the concept of desktop to teraflop as a goal for their systems architectures and software environments. Without doubt, much is riding on the assumption that the community can develop the insights and the methods for building scalable systems software and environments and turn

these ideas into working software that will have broad deployment and impact. It is therefore critical that the community have at least one testbed that is focused on providing the needed infrastructure for this type of research.

The expected impact of this project is considerable but will be achieved only if the community can exploit the capabilities of the testbed on a routine basis. We believe that a dedicated testbed is required to support this important work. The testbed must be available to the computer scientists for routine use without having to move applications off the machine, so that ideas can be tried while they are still fresh and while feedback can be taken into account into the design process. The testbed also must be organized in a way that users can have root access to the compute nodes and can try out low-level software environments on a routine basis without risking the instability of the whole system. The proposed future Chiba City testbeds will be designed to support quick reconfiguration of the nodes and software to enable users to have root privileges without incurring security risks or undue downtime for system recovery between experiments.

3.1 Scalability

As mentioned in the Introduction, Little's law tells us that to build systems with ever-increasing performance, we must address increasing amounts of concurrency. This concurrency may reveal itself in many ways in real systems, through increasing the number of nodes in the system, the numbers of threads supported in hardware, the number of memory channels or the number of functional units. Modern microprocessor microarchitectures have endeavored to keep pace with increasing performance demands by increasing the number of functional units that can be executing in parallel via instruction-level parallelism and in the near future with simultaneous multithreading. Vendors have explored new CPU/Memory interface technologies such as RAMBUS and DDR to improve memory bandwidth and concurrency and they have begun to exploit chip based node parallelism through multiple cores per die. All of these mechanisms have resulted in improved performance at the CPU and node level.

Improving performance at the system level, however, will require more nodes. Current estimates indicate that petascale systems will require levels of concurrency of order 1-10 million, with conservative estimates requiring systems with CPU counts of order 100,000. Some more experimental designs have node or CPU counts of order 1,000,000. Commodity cluster-based designs can and probably will be built with 10,000 nodes during the next three to five years. It is not clear, however, whether current commodity-based designs are practical much beyond this 10,000-node range, in terms of hardware reliability and serviceability.

Nevertheless, regardless of design, future large-scale HPC systems will ─ by physical necessity ─ support unprecedented degrees of concurrency, much of which will be visible to both the operating system and the user. It is this last point that is the target of this proposal. Simply put─*the HPC community will be unable to address the software requirements of these future systems in a timely manner unless researchers have the testbed resources and software environments to explore these issues.* We have designed the sequence of Chiba City Project testbeds to address this important current and ongoing need.

Today's most productive cluster systems are those with 128 or fewer single, dual, or quad-processor nodes. This scale allows for the highest-performance interconnect technologies and the most stable software environments. Larger node counts, such as Argonne's 256-node Chiba City cluster, press the edges of scalability. Significant I/O bottlenecks begin to appear at this point,

and maintaining the stability of the systems software infrastructure (such as parallel file systems and schedulers) is difficult. Some groups have radically simplified the environments on nodes to cope with this scalability by running the nodes diskless or with reduced services. While this may be appropriate for some applications classes, it is not clear that it is the best or only way to achieve future scalability requirements. Our view is the 1000-node range represents the current state of the art in clusters, and experience beyond this is largely unproven, with at best limited success.

The DOE ASCI program has perhaps the most experience with high-node count machines (Red, Blue Pacific, Blue Mountain, White, and future machines Q and Purple). This experience indicates that it is nontrivial to scale systems software and applications to these machines without substantial amounts of testing and development. The key difference between the ASCI systems and the clusters being built by the community is vendor-integrated systems with proprietary operating systems and non-commodity processors. This difference only magnifies the difficulty of addressing high-levels of scalability due to the lack of critical mass of developers and access to dedicated testbed resources. IBM, Compaq, SGI and HP all have stated a strong interest in participating in the Chiba City Project and all have substantial internal efforts to align future scalable systems platforms with the open source community in part to obtain more critical mass of developers all aimed at improved scalability in the future. IBM for example is developing a native 64-bit Linux kernel for Power4 and Intel/SGI and HP have been collaborating on the IA-64 Linux environment. It is their openly stated belief that progress on scalability in the cluster realm and open source systems software environments can be directly translated into improved scalability for proprietary systems and in some cases may simply displace proprietary work in this area when they can incorporate the open source solutions directly in their product line.

The NSF Distributed Terascale Facility will be based entirely on Linux and IA-64 systems with the largest single system planned to support about 1500 CPUs on about 700 nodes. The DTF systems will be available for very limited testing of new systems software, as they will be a dedicated production resource. However their existence simply points out the growing importance to support existing systems software packages on systems of order 1000 nodes.

While current DOE Office of Science large-scale production computing platforms are based on IBM SP class machines (NERSC-3 and ORNL's Power4 system) it is quite reasonable to assume that future large-scale production machines at NERSC and ORNL could (and perhaps should) be based on open source scalable cluster technology. Cluster based systems would provide a smooth scalability path for those researchers that already are computing on smaller clusters in their own labs and as a result may offer substantial price-performance benefits to the DOE program. These systems will be operated as a production resource and will likely target performance in the 10-20 Teraflops range in the next few years. These performance targets imply systems in excess of 1000 CPUs and perhaps greater than 1000 nodes depending on the level of node integration. The Chiba City Project could have substantial impact on enabling the feasibility of clusters for future Office of Science needs.

Finally, the aggressive research machines the community is looking at (e.g., Blue Gene/X) have considerable unresolved scalability issues and those issues are active topics of research. We believe that the Chiba City II testbed and future systems could be used by some of these communities for a class of studies that might be prove important to future architecture development, in particular exploring issues relating to fault recovery and configuration management in the face of unreliability.

To tackle high node-count problems without purchasing an immense (and therefore expensive) system, we plan to exploit the combination of low-cost commodity microprocessor technology and virtual Linux node software in order to build a "virtual subsystem." The first proposed system will have 1K physical nodes and allow simulation of an 8K-node cluster. The second will have 2K physical nodes and 40K virtual nodes. The final proposed system will have 4K physical nodes and as many as 160K virtual nodes. Such a capability will also provide for testing various granularities of node processor-count, using *n* virtual nodes per physical node to simulate *n*-processor nodes.

The testbed will be a resource to computer scientists and developers who need to develop and test their software at real scale on a real computer. On large computers, the majority of software (including the systems software, numerical libraries, and application frameworks) is never seriously tested at scale because developers usually do not have access to a large system for development and testing. The current Chiba City cluster is unique in that it is specifically engineered and run for computer science research, but it is aging and only of moderate size (256 compute nodes).

The Chiba City II machine will be configured explicitly to support the development and testing of systems software that needs to run as root (i.e., it will have additional control nodes and filesystems that support quick recovery of a standard configuration) without incurring a security risk. In addition, the system will be configured to permit the collection of low-level performance data of interest to the systems software developers. Larger clusters are available in a couple national laboratories (e.g. C-plant), but the salient difference is that these clusters are operated in support of "production" applications. As a result, software developers and computer scientists are able to address issues only at the application layer, and perhaps at the messaging layer, but not at the systems level (i.e., alternative kernels, filesystems, schedulers). This restriction presents a significant problem because it is precisely at the systems level that scalability issues lie. Indeed, the current lack of scalability of the systems-level hardware and software components, such as interconnect technologies and parallel file systems, presents the most fundamental barrier to reaching the scale of cluster required to move beyond tens of teraflops, much less reach into the petaflops regime. Additionally, the Chiba City testbed will support new avenues of research into scalability issues. In much the same way that new laboratory equipment can enable different experiments and bring about new results, this cluster will enable researchers to ask questions and develop solutions that may not otherwise have been possible.

3.2 Extreme Development

The majority of deployed clusters are production systems that are used to support computational science applications and simulations. In fact, all large clusters available to HPC scientists are focused on providing production-quality computing cycles. One type of software development, however, is not well suited for production systems, because it introduces instabilities into the system that can affect other users or even crash the computing system. We call this type "extreme development." Extreme development is often necessary in computer science, in systems software development, and even in some scientific application programming. These activities directly affect the machine's computing environment and are fairly likely to cause problems for the users of the system, particularly during testing and debugging. For example, effective testing may require the replacement of critical components of the systems software, such as file systems, low-level communication drivers, or process management systems, with experimental (and unstable) versions. And, when developing software that must be fault tolerant, actual failures of

hardware and systems software must be introduced in order to test the software fully. Conducting these experiments on a production system is impractical at best and impossible at worst. Ironically, it is this very type of activity that eventually enables large production systems to be built. Some mechanism must exist in order to test the software that will eventually become part of a production system.

To address this problem, we include in the proposed systems a "reliability subsystem," a concept that was originally tested on the existing Chiba City. The reliability subsystem includes remote power control and monitoring, automated console management, and a system of rock-solid management nodes invisible to the users of the cluster. Using the reliability subsystem, one can rebuild a computing node or I/O node entirely from scratch with no operator intervention. Users can have root (administrative) access on their nodes with no specific concerns, and nodes can operate with effectively any operating system. Soft failures of individual cluster components can be introduced, and hardware components can be powered off in order to simulate real hardware failures. With these capabilities of the reliability subsystem, it doesn't matter what a developer does to a node during testing and debugging—the system can recover painlessly. Key advances in the scalability of the Parallel Virtual File System (PVFS), Myrinet networking, and process management were made by using the Chiba City cluster in this mode.

3.3 Requirements Analysis

These two goals – supporting scalability and supporting extreme development – go hand in hand. Indeed, many projects need both at the same time. As a part of the planning for these systems, we spoke to many scientists from all areas of the potential user community for these clusters. We wanted to ensure that all of the requirements for their various projects could be met, and to develop a better overall understanding of the needs. A representative summary of the requirements in various areas is given in the following table.

	Scalability			*Extreme Development*			
	Kilo-CPUs	**Kilo-images**	**Kilo-nodes**	**Root Access**	**Specialized Kernels**	**Dynamic Node Software**	**Dynamic System services**
Schedulers and Resource Mgrs	X	X	X	X	X	X	X
Programming Models	X		X		X	X	
Kernel Development				X	X	X	X
File System Development	X	X	X	X	X		X
Communication Libraries	X		X		X	X	X
Networking Hardware			X	X	X	X	X
Debuggers	X	X	X		X	X	

The project types listed on the left of the table include many of the active areas of research that will use the system; each represents many specific projects. For example, "Schedulers and Resource Mgrs" summarizes the requirements of many of the projects in the Scalable Systems Software SciDAC ISIC, such as MPD and the Maui scheduler. It also represents Veridian's PBS Pro and Platform's LSF, as well as scheduling research projects at Northwestern University and the University of Wisconsin.

In this table, an "X" in a box indicates the following:

Kilo-CPUs	Projects in this area require 1024 or more CPUs.
Kilo-images	Projects in this area require 1024 or more separate system images.
Kilo-nodes	Projects in this area require 1024 or more separate networked nodes.
Root Access	The developers of these projects require privileged access to the nodes.
Specialized Kernels	This type of project requires a custom kernel to be installed.
Dynamic Node Software	This type of project requires frequent and/or unstable changes to the node software.
Dynamic System Services	This type of project may require frequent and/or unstable changes to the system infrastructure (such as naming services, mapping services, file systems).

4 Approach

While this proposal is primarily focused on a specific machine for FY02 installation, we believe that the community needs access to a series of machines during the next five years that will keep pace with the requirements for commodity cluster scalability. These must also segue to the needs of next-generation systems that will be based on future commodity processor cores but with more efficient packaging and integration (i.e., future commodity-based petaflops systems based on cluster-on-a-chip technologies that are likely to follow the current system-on-a-chip trend).

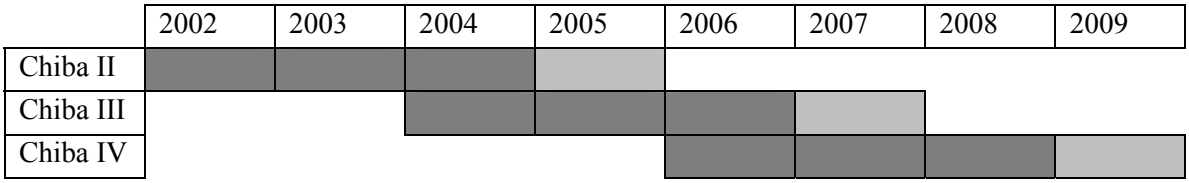
At the highest level, we are proposing a testbed consisting of three cluster installations over a five-year timeframe. The deployment schedule is based on an analysis of Moore's law and commodity hardware and takes into account the time required to install a cluster and the useful lifetime of commodity systems.

The approach for each individual system is relatively simple:

1. Build a cluster based on commodity hardware.
2. Bring up the virtual subsystem, and characterize it.
3. Operate the system as a user testbed.
4. Perform exploratory tests on emerging hardware to prepare for the next system.

4.1 System Timeline

The timeline that we propose is shown here:



This schedule includes these considerations:

- A large cluster takes, on average, six months to be fully installed and debugged, pass acceptance tests, and become available to the entire user community.
- A cluster's expected life is about three years, with an extra year of uptime to allow projects to migrate. ("End of life" of a system is shown in light gray.)
- The second cluster will be installed toward the end of the lifetime of the first cluster, such that the first cluster can continue to be used while the second cluster is in "early user" mode. This strategy gives projects sufficient time to move to the new system while still making progress on the old. The same is true for the second and third clusters.
- The third system will most likely be reaching its end of life when petaflops systems are reaching full production. Thus the work enabled on the third cluster will have direct impact on early petaflops systems.

4.2 System Configurations

A second compelling motivation for installing three clusters, each two years apart, is to take advantage of Moore's law in order to steadily increase the size of the virtual subsystem. The following chart shows the expected growth in CPU power, RAM, network bandwidth, and storage per node, where the price of the node remains constant over time. These numbers may be off by a factor of up to 10% in either direction (due, in part, to the jump of commodity systems from 32-bit to 64-bit architectures during this timeframe), but the numbers give a good sense of the basic capabilities. Because the footprint of a virtual Linux system remains constant at 128M while the amount of RAM/CPU increases, we can achieve approximately 160K virtual nodes by 2006.

	Chiba I	**Chiba II**	**Chiba III**	**Chiba IV**
Year	1999	2002	2004	2006
FLOPs/CPU	500 MF	2 GF	3.5 GF	6.3 GF
RAM/CPU	256 MB	1 GB	2.5 GB	5 GB
Network BW	100 MB	200 MB	500 MB	1 GB
Storage/node	9 GB	100 MB	400 GB	1 TB
Nodes	256	1024	2048	4096
CPUs	512	1024	2048	4096
Virtual Nodes	256	8192	40960	163,840

4.2.1 System Details for the 2002 Cluster

The 2002 cluster (Chiba City II), will include the following system components:

- 1024 user nodes. Each will have a single 2.4 GHz Pentium IV, 1 GB of RAM, and 100 GB of local disk. The nodes will be single processor CPUs in order to maximize the number of physical nodes and OS images (as opposed to maximizing CPUs by purchasing SMP systems). The 1 GB of RAM will support 8 virtual systems. The local disk is large enough to support I/O experiments.
- Myrinet 2000 for high-speed interconnect between user nodes. We have selected Myrinet because it has the highest bandwidth, lowest latency, and best scalability for the cost.
- Fast Ethernet for the management network and reliability subsystem.
- 18 storage nodes. These will have dual 2.4 GHz Pentium IVs, 1 GB of RAM, and 1 TB of disk. Two of these nodes will be for dedicated nonvolatile user file storage, inaccessible to potential side effects of experiments. The remaining sixteen will support I/O, data, and file system experiments on the cluster and will be accessible to user projects.
- 18 management nodes, identical to the storage nodes. These make up the reliability subsystem. Sixteen of these of these will for configuration management and infrastructure services for the user nodes. The other two will be used for central management of the entire cluster.
- Monitoring and management hardware. Included are remote power systems, remote console systems, and remote status monitoring hardware.

Some of the projects planning to use this testbed have suggested that it would be useful to configure a subset of the user nodes, perhaps thirty-two or sixty-four, as dual-CPU systems. This would support both kernel work, which needs to test CPU locking, and certain programming models. The decision as to whether or not to do this will be based on financial feasibility at the time of the purchase and the possibility that other projects may provide this capability. The software environment for the cluster will be built on Linux as the primary OS. The management nodes will always use Linux and a mix of community codes in order to support the cluster. The user nodes will, by default, run Linux and the standard set of compilers and development tools commonly available on Linux clusters today (PGI compilers, MPICH, Totalview, etc). User nodes also will have the capability to run any other OS or development tools selected by the user, subject to portability and financial restrictions. For example, at any particular time, a set of nodes might be running in the Scyld cluster model or using FreeBSD as a node OS.

4.2.2 System Details for Subsequent Clusters

It is too early to predict the exact system details for the clusters in the 2004 and 2006 timeframes. Based on Moore's law, we can anticipate that these systems will have approximately 2048 and 4096 user nodes, respectively. The network interconnect choice for these systems will be interesting. We expect to see major changes in options for the interconnect based on the growing prevalence and decrease in cost of gigabit Ethernet, the deployment of Infiniband, and the evolution of Myrinet. All of these activities give us confidence that the bandwidth goals for the system, as shown in the chart above, will be met.

4.3 Virtual Subsystem Approach

In order to maximize the capability to test at scale, we have developed the notion of a "virtual subsystem" of a cluster. One real user node will be configured to run several virtual user nodes, with the exact number dependent on the amount of RAM on the user node and the needs of the project on the system at the time. A virtual user node will run a complete copy of the standard cluster configuration, including Linux, third-party software, file systems, and network access.

Such virtual nodes can be created by using VMware (a commercial product) and User Space Linux (an open source product). Our testing of these products reveals that both will support standard user node capabilities. Neither has native support for Myrinet at this time, but this appears to be a solvable problem and we will work with Myricom to address this. In the interim, the Myrinet network is accessible over IP to the virtual node. Both require approximately 128 MB of physical RAM as a footprint; thus each node on the 2002 cluster will be able to support 8 simultaneous virtual nodes.

The virtual subsystem will be available to users to schedule and operate on much as if they were standard physical nodes. This capability will, for example, allow a scheduler or a file system to be tested on 8000 nodes—something that is effectively impossible today. We emphasize that these tests will not provide particularly useful information about performance—this is not the goal. Rather, the goal is to be able to test capability and locate unintentional bottlenecks and barriers in algorithms and implementation. However, it will be possible to roughly predict performance by running an experiment on the same number of real nodes and virtual nodes. For example, the results of an I/O test on 800 real nodes can be compared with the results of a test on 800 virtual nodes (i.e., 100 real nodes each running 8 virtual nodes). Such a comparison would measure, in part, how much impact the virtual subsystem was having on performance.

4.4 User Support and Outreach

Much like the existing Chiba City, the testbed systems will be operated in the following way:

- Priority access to the system will be given to those projects that test or develop at scale, carry out scalability research, or require support for extreme development.
- Any remaining time on the system will be allocated to computational science applications, which will be chosen based on their ability to exercise the various projects using the cluster as a testbed. It is likely that experimental file systems, resource managers, or libraries will be a part of the cluster, and those applications that can most effectively test those will be selected.
- We will implement an allocation process as approved by MICS. Details of the allocation decisions (both in terms of basic access and total amount of time) will be made by the MCS computing committee, with priorities based on the parameters described in this proposal and as determined by MICS.
- The projects will be able to reserve as much as the entire cluster for significant periods of time, up to a week, in order to perform intensive testing and analysis at maximum size.

It will be important to foster an active user community of the systems, both to ensure that the support of the system is as responsive as possible and to disseminate early research findings. We will carry out the following activities:

- Holding workshops on scalability research and development, open to the research community.
- Arranging meetings of users and developers of the systems software.
- Collecting annual progress updates from all projects on the system.
- Soliciting input from the key projects on the system as a part of the technical specification of the 2004 and 2006 clusters.

5 Budget

The estimated budget for the testbed is as follows:

Year	Items	Operations	Hardware
FY2002	1024 node cluster + Operations	$1.0M	$5.0M
FY2003	Operations	$1.0M	
FY2004	2048 node cluster + Operations	$1.0M	$5.5M
FY2005	Operations	$1.1M	
FY2006	4096 node cluster + Operations	$1.1M	$6.0M

The budget for the FY2002 cluster is based on specific quotes acquired from reliable vendors and assumes a substantial discount. The exact dollar figures from the quotes are not included directly because the figures change quickly in the commodity space. The clusters in FY2004 and 2006 are based on an extrapolation of costs from today's system rather than from real quotes. No vendor in the commodity market will project prices out 2-3 years. Funding for "Operations", i.e. personnel and infrastructure costs, are in a separate proposal.

6 Technical Progress

This testbed is a logical continuation of work already taking place at Argonne. In 1999, we built Chiba City, a 256-node Linux cluster to support a more modest version of these same goals. As a part of the Chiba City Project, we have investigated scalability techniques, developed tools for supporting extreme development, facilitated critical efforts in the cluster community, and published observations and results.

Notable among these accomplishments are the following:

- The development of the first instantiation of the reliability subsystem, built around the concept of cluster "towns," "mayors," and remote control mechanisms. Nodes on Chiba City can be remotely recovered from crashes and rebuilt with any operating system image without user or administrator intervention. This capability allows users to have root access on their nodes, experiment with kernels, and change node configurations.
- The release of the City Toolkit, which includes tools for large-scale cluster management.
- Active support of a number of research projects that have used the entire cluster to push scalability as far as possible on the system, including MPICH and MPD, PVFS, MM5 climate code, and quantum physics.
- Dedicated support and scalability testing of the Scyld BProc cluster software and Myrinet's GM libraries. Both of these activities resulted in changes to the software released by these companies, changes that in turn benefited other cluster efforts.

Furthermore, we have operated large computers in support of user communities focused on research for well over ten years. In the mid-1980s and early 1990s, Argonne operated the Advanced Computing Research Facility, which included a diverse collection of parallel machines. As a part of the ACRF, we regularly taught parallel computing techniques through a series of week-long courses. We will follow a similar approach as part of the scalability outreach

program. Since that time, we have fielded a number of IBM SPs, a large SGI Origin 2000, and many different flavors of clusters.

7 Relationship to Other Efforts

The proposed Chiba City Project builds on the experiences gleaned in our first scalability testbed, Chiba City I. On Chiba City I, we developed a robust mechanism for supporting extreme development, learned how to handle the diverse needs of computer scientists, and began to make progress on the scalability research frontier. In comparison with Chiba City I, the follow-on testbeds will be larger, will support the ability to expand beyond the physical node count by using a virtual subsystem, and will be available to a much larger user community. The expanded Chiba City scalability testbeds will provide an invaluable resource for computer scientists and application developers in many DOE-funded research projects, including the DOE SciDAC programs and existing DOE-funded base program research activities. Moreover, the Chiba City scalability testbeds will expand the scope and mission of research enabled by existing Advanced Computing Research Testbeds and MICS-funded Research Facilities. Because of the project's focus on scalable computer science and extreme development and its accessibility to all such projects, the testbeds will be a unique resource that enhances the capabilities of DOE's entire research portfolio.

The Chiba City Project testbeds will support the research and development components of an extensive number of projects over time. In the near term, these projects will include:

- The Scalable Systems Software Center. This project is a SciDAC Integrated Software Infrastructure Center coordinated out of Oak Ridge National Laboratory. It will do the research for and produce an integrated suite of systems software and tools for the cost effective management and utilization of terascale computational resources.
- The Scientific Data Management Center. This is a SciDAC Integrated Software Infrastructure Center coordinated out of Lawrence Berkeley National Laboratory. The center's goal is to provide a coordinated framework for the unification, development, deployment, and reuse of scientific data management software.
- The Performance Evaluation Research Center. This is also a SciDAC Integrated Software Infrastructure Center, which is coordinated out of Lawrence Berkeley National Laboratory. The Center is developing a science for understanding performance of scientific applications on high-end computer systems, and is also developing engineering strategies for improving performance on these systems.
- The Science Appliance Project. This project, which is led by Los Alamos National Laboratory, explores alternative approaches to cluster design, management, and system software in order to build particularly simple and effective clusters.
- The Sandia Micro kernel Project. We have begun discussions with researchers at SNL about their needs for software development testbeds that complement the resources they have available at Sandia. Chiba II could be a critical enabling resource for this activity.
- DOE Base program projects such as future programming models project, MPICH, ACTS toolkits and parallel I/O will benefit from sustained access to the Chiba II testbed in the same fashion they have benefited from access to the Chiba I system. In fact much of the development work and testing would not be possible without such testbeds.
- DOE projects in scalable scientific visualization will also benefit from access to Chiba II for testing parallel rendering algorithms and for validating large-scale data management strategies.

- Early Career Principal Investigators—Finally we plan to make the Chiba II system available to DOE Early Career Principal Investigators from universities. For many researchers, access to a scalability testbed will be a critical resource to enable them to contribute ideas and software without the difficulty and overhead of attempting to build their own testbeds. This will encourage EC PIs to collaborate with and contribute to SciDAC projects.

Select Publications

R. Butler, W. Gropp, and E. Lusk, "Components and Interfaces of a Process Management System for Parallel Programs," Parallel Computing 27 (2001), pp. 1417-1429

R. Butler, W. Gropp, and E. Lusk, "A Scalable Process-Management Environment for Parallel Programs," Preprint ANL/MCS-P812-0400, April 2000.

P. H. Carns, W. B. Ligon III, R. B. Ross, and R. Thakur, "PVFS: A Parallel File System for Linux Clusters," Proc. Extreme Linux Track: 4th Annual Linux Showcase and Conference, Oct. 2000.

P. M. Dickens and R. Thakur, "On Implementing High-Performance Collective I/O," Preprint ANL/MCS-P852-0900, Sept. 2000.

R. Evard, "Chiba City: A Case Study of a Large Linux Cluster", in Beowulf Cluster Computing with Linux, by Thomas Sterling. MIT Press, 2001.

W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith, "Performance Modeling and Tuning of an Unstructured Mesh CFD Application," Preprint ANL/MCS-P833-0700, July 2000.

S. C. Pieper, V. R. Pandharipande, R. B. Wiringa, and J. Carlson, "Realistic Models of Pion-Exchange Three-Nucleon Interactions," Phy. Rev. C 64, 01400101-21

R. Ross, D. Nurmi, A. Cheng, M. Zingale, "A Case Study in Application I/O on Linux Clusters," Preprint ANL/MCS-P888-0701, July 2001.

R. B. Ross and W. B. Ligon III, "Server-Side Scheduling in Cluster Parallel I/O Systems," Preprint ANL/MCS-P915-1101, Nov. 2001

J. Taylor and J. Larson, Resolution Dependence in Modeling Extreme Weather Events," *Proc. 2001 Int'l conf. on Computational Science*, eds. V. N. Alexandrov, J. J. Dongarra, and C. J. K. Tan, Springer-Verlag (to appear).

J. Taylor, J. Larson, and S. Voeltz, "Climate Modeling at the Regional Scale," Climate and Global Change Report ANL/CGC-009-0601